MIDTERM EXAM II SOLUTIONS
CSCI 61: DATA STRUCTURES
SPRING 2016

1. (10 points) Simulate step-by-step the function `build_heap()` on the following array:

   `9 1 8 3 7 6 5 2 10 4`

   Show the array after each swap.

   Answer:

   ```
   9 1 8 10 7 6 5 2 3 4
   9 10 8 1 7 6 5 2 3 4
   9 10 8 3 7 6 5 2 1 4
   10 9 8 3 7 6 5 2 1 4
   ```

2. (10 points) Let T be an empty open addressing hash table with linear probing of size 7, and let its hash function be $h(n) = n \% 7$.

   Carry out the following table operations in the given order and show the table after each operation:

   `insert 27, insert 5, insert 6, insert 40, delete 27, insert 12, insert 13, delete 12, insert 33, insert 12.`

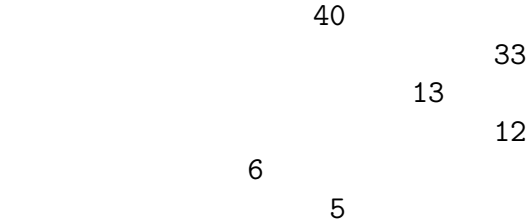
   Answer:

```
0: 6
1: 40
2: 13
3: 12
4:
5: 5
6: 33
```

3. (10 points) Let T be an empty binary search tree. Carry out the following tree operations in the given order and show the tree after each operation:

   `insert 27, insert 5, insert 6, insert 40, delete 27, insert 12, insert 13, delete 12, insert 33, insert 12.`

   Answer:



4. (10 points) What is the output of the following program:

```
#include <iostream>
#include <queue>
#include "btnode.h"

using namespace std;

int main()
{
  int a[] {5, 1, 2, 0, 4, 6, 9, 7, 8, 3};
  btnode<int> * root(nullptr);
  for (auto e: a)
    bst_insert(root, e);

  queue<btnode<int> *> q;
  q.push(root);
  while (!q.empty())
    {
      btnode<int> *ptr = q.front();
      q.pop();
      if (ptr != nullptr)
        {
          cout << ptr->data() << endl;
          q.push(ptr->left());
          q.push(ptr->right());
        }
    }
  return 0;
}
```

Answer:

```
5
1
6
0
2
9
4
7
3
8
```

5. (10 points) A **reverse** binary search tree is a binary tree in which each node is larger than the nodes in its right subtree and smaller than the nodes in its left subtree (assume there are no duplicates).

   Write a recursive template function `void bst_reverse(btnode<T> * & root_ptr)` to convert a binary search tree to a reverse binary search tree. The shapes of the old and new trees should be mirror images.

   Answer:

```
template <class T>
void bst_reverse(btnode<T> * & root_ptr)
{
    if (root_ptr == nullptr)
        return;

    bst_reverse(root_ptr->left());
    bst_reverse(root_ptr->right());
    std::swap(root_ptr->left(), root_ptr->right());
}
```